



Abstract Reviewed Paper at ICSA 2019

Presented * by VDT.

Deep Neural Network Approaches for Selective Hearing based on Spatial Data Simulation

S. Hestermann, H. Lukashevich, C. Sladeczek
Fraunhofer Institute for Digital Media Technology IDMT

Abstract

Selective Hearing (SH) refers to the listener's attention to specific sound sources of interest in their auditory scene. Achieving SH through computational means involves detection, classification, separation, localization and enhancement of sound sources. Deep neural networks (DNNs) have been shown to perform these tasks in a robust and time-efficient manner. A promising application of SH are intelligent noise-cancelling headphones, where sound sources of interest, such as warning signals, sirens or speech, are extracted from a given auditory scene and conveyed to the user, whilst the rest of the auditory scene remains inaudible. For this purpose, existing noise cancellation approaches need to be combined with machine learning techniques. In this context, we evaluate a convolutional neural network (CNN) architecture and a long short-term memory (LSTM) architecture for the detection and separation of sirens. In addition, we propose a data simulation approach for generating different sound environments for a virtual pair of headphone microphones. The Fraunhofer SpatialSound Wave technology is used for a realistic evaluation of the trained models. For the evaluation, a three-dimensional acoustic scene is simulated via the object-based audio approach.

1. Introduction

Conventional closed-back headphones block environmental sounds through insulated ear cups. Noise-cancelling headphones, on the other hand, use on-board processing to cancel ambient sounds through destructive interference [8]. Under certain conditions, this technology poses one significant problem. Since most algorithms rely on basic physical principles, they lack semantic understanding of the canceled signal. Consequently, any sound is blocked, regardless of its potential importance to the headphone user. Information- and time-critical sounds, such as sirens in traffic, thus may not receive the user's attention and provoke dangerous situations.

This issue may be solved through source separation on the ambient audio stream from the integrated headphone microphones. Sirens as an example for critical sound sources may be isolated from the auditory scene and played back on the headphones. This falls into the category of Selective

Hearing (SH) which has seen many advancements in terms of detection, classification, separation, localization and enhancement of sound sources [1].

Existing DSP-based source separation approaches that may be used for SH applications are commonly based on statistical means. The common goal is the estimation of the inverse of the mixing matrix A which was used to mix N real source vectors $s(t) = (s_1(t), \dots, s_N(t))^T$ to M output vectors $x(t) = (x_1(t), \dots, x_M(t))^T$:

$$x(t) = A s(t) . \quad (1)$$

Since both A and $s(t)$ are unknown, finding A^{-1} is an ill-posed problem. Furthermore, in the context of practical applications $M \ll N$. The solution to this problem is therefore approximated under various assumptions [2].

Independent component analysis is a statistical source separation method under the assumption that the sources are

statistically independent and identically, but non-Gaussian distributed [2]. A further common separation approach is non-negative matrix factorization which assumes that the mixing matrix A and the sources s are non-negative [2]. Regarding more recent approaches, projection-based demixing was introduced [4]. In this approach, the observed output mixture $x(t)$ is decoded into tensors of time, frequency and channels. Then, different spatial projections are created within these tensors to identify individual sources.

In contrast to the previous methods for DSP-based source separation, the use of a neural network architecture poses a more promising application-focused approach for the separation of sirens in particular. Since sirens are usually simple combinations of sinusoidal sounds, neural networks can be trained to detect and separate them from a diffuse sound ambience. This poses no further constraints on the audio material except a certain level of presence of the siren, which may be negligible within everyday boundaries.

In this context, we compare a CNN and LSTM model embedded into a corresponding system architecture for the extraction of sirens from a stereo time signal, without digital signal processing requirements or assumptions. The data preparation step simulates signals as they may arrive at microphones integrated into headphones. Apart from accuracy metrics, the proposed system is tested in a simulated traffic scenario. The acoustic scene is generated via the object-based audio approach of the Fraunhofer SpatialSound Wave technology [5].

2. Data Collection

A diverse set of training and validation data was collected for training and evaluation of the DNN models. About 200 Gigabytes of online audio material were scanned for the curation of two final datasets of 20 Gigabytes in total size. One dataset was used for training, the other one for testing. In addition to the evaluated online content, free field recordings were made to further expand the dataset. The audio material was sampled at 44.1 kHz and a resolution of 24 bit.

The training and test dataset are comprised of two parts: ambience sounds and siren sounds. The curated ambience sounds mostly consist of various traffic ambience recordings, as these represent the setting where siren sounds most likely occur. Apart from traffic recordings, other typical city and outdoor ambiences were included, such as from parks, malls, train stations and similar public places. A minor part of the curated ambience sounds were obtained from the UrbanSound data set [14]. In order to challenge the DNN models, white noise and pink noise sequences, as well as ambience recordings with sounds present in a frequency range similar to sirens were added, e.g. twittering birds and playing children.

The majority of the curated siren sounds are free from strong artificial or recorded reverb as well as delay effects to ensure network model training without data pollution. For further data cleansing, the siren sounds were high and low cut at 150 Hz and 7500 Hz, respectively, in order to remove low frequency rumble and irrelevant high pitched noise.

3. Spatial Data Simulation

Signals as they may arrive at microphones placed on the outside of ear cups were simulated using a custom acoustic simulation script. This allowed for automated time delay and distance dependent gain calculations using two virtual microphones, as shown in Figure 1.

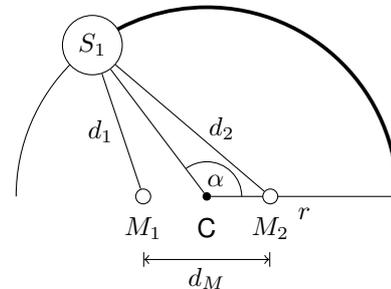


Fig. 1: Virtual microphone simulation. The position of the sound source S_1 is defined by the radius r and the azimuth α . Depending on the distances d_1 and d_2 between S_1 and the microphones M_1 and M_2 , the gain and time delay for each microphone signal is calculated.

The simulation works as follows. Two virtual microphones M_1 and M_2 are placed at a distance d_M from one another, centered around C . The position of a virtual sound source of interest S_1 is defined by the radius r and the azimuth α , alongside its initial gain $g(S_1)$. The gain of the S_1 signal arriving at M_1 and M_2 is then calculated using the distances d_1 and d_2 between the microphones and S_1 via equation 2:

$$g(S_1, M_{1|2}) = \frac{g(S_1)}{d_{1|2}^2}. \quad (2)$$

Additionally, the time delay Δt for the signal from S_1 arriving at each microphone is calculated using equation 3 with the speed of sound v_s at 343 m/s:

$$\Delta t(S_1, M_{1|2}) = \frac{d_{1|2}}{v_s}. \quad (3)$$

The stereo mix of the mono microphone signals creates the impression of the sound source of interest coming from the intended position. The distance d_M can be adapted to the outer distance between the ear cups of different headphones. The simulation may be improved in the future through the simulation of the human head, since with this setup a distance $d_M \geq 1$ m led to the most realistic simulations [12].

4. Neural Network Models

A CNN and LSTM model were evaluated for the task of reliable separation of sirens. The models were implemented using the Keras API of Google Tensorflow [3] and embedded into the corresponding system architecture presented in section 5.

The proposed CNN model is based on the model introduced in [10]. It is fed with 25 chronological STFT windows in order to predict an STFT mask for the central 13th STFT window which is then used to separate the source of interest.

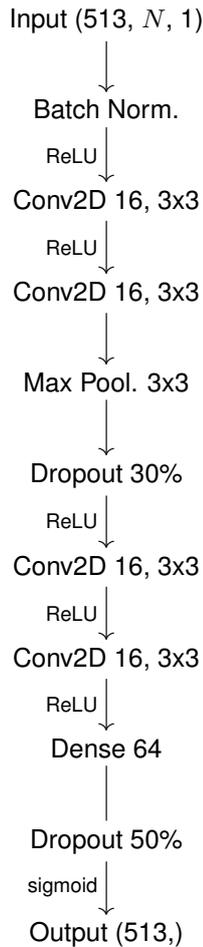


Fig. 2: Spectrum masking model. The model is fed with packages of N STFT windows. After batch normalization, one output STFT window mask is predicted through multiple convolutional, dropout, dense, and one max pooling layer.

The original model from [10] was optimized in several steps, resulting in the final model shown in Figure 2. In comparison to the model in [10], the leaky ReLU activation functions were substituted by regular ReLU activations [16]. The dropout rates were increased to 30% to counteract model overfitting [15]. Apart from increased dropout rates, a batch normalization layer was added to normalize the input before the convolutional layers [7]. This yielded better model predictions on a larger variety of input data during early testing. From a practical standpoint, this also produces a more constant output volume of the separated sound source.

In a last step, the stochastic gradient descent optimizer of the original model with custom parameters was replaced by the Adam optimizer [9]. This decreased training duration and further improved model performance. Due to exploding gradients, a clip value of 3.0 was set for the optimizer [13].

The input of the network as denoted in Figure 2 has the two-dimensional shape $513 \times N \times 1$, where the 513 frequency bins result from the STFT window size of 1024 samples, as further explained in section 5. The N windows correspond to the 25 chronological STFT windows of the original model from [10]. This time context of $N = 25$ was changed to

$N = 17$ windows in order to evaluate network predictions with less data. These results are presented in section 6.1.

For potential performance improvement, the chronological context of each predicted STFT window was also replicated by an LSTM model [6]. The tested LSTM model uses all layers of the CNN model in Figure 2 before the output layer as time distributed input to an LSTM layer with 64 LSTM cells. The metrics of this alternative model are also discussed in section 6.1. Despite significantly longer training times, no practical prediction improvements compared to the CNN model were identified.

5. System Architecture

The complete system architecture for the separation of sirens is depicted in Figure 3. One instance of the DNN is applied

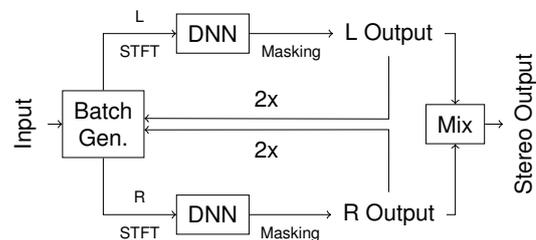


Fig. 3: Stereo separation pipeline. The stereo ambience mix is split into left and right channels by the batch generator. The DNN for separation is then applied three times to each channel individually. The mono output of the two network instances is mixed back together to obtain the final stereo output.

to each channel of the stereo input signal. The correct localization of the siren signal is preserved through the final mix of the two separated mono signals.

As presented in the previous section, the CNN and LSTM models operate on data in the frequency domain in order to learn spectrum masks. These masks are applied to the STFT windows of the mixed ambience input, which separates the siren by isolating corresponding frequency bins. The final processing pipeline for this approach is depicted in Figure 4. The network is trained with a processed ambience mix, and

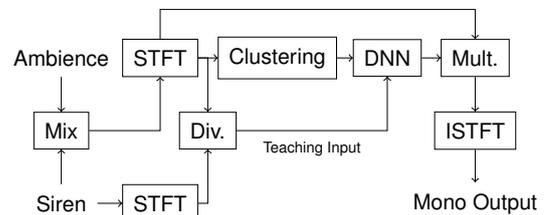


Fig. 4: Spectrum masking pipeline. The ambience mix and siren signal are converted to the frequency domain. The DNN is trained to predict spectrum masks to separate the siren signal from the mono ambience mix.

a processed version of the siren signal as the teaching input. Both the mix and the siren signal are first transformed into the frequency domain via STFT. The transformation uses the Hann window function on a window size of 1024 samples and

a window overlap of 512 samples. This results in complex STFT windows in the shape 1024×1 .

Since the neural network only operates on real data, the magnitude spectrum of each STFT windows is used for further processing. This results in spectrum windows in the shape 513×1 . For the teaching input of the network, the magnitude windows of the ambience mix and the siren signal are divided by one another to obtain the desired spectrum masks. The calculation of each spectrum mask \vec{m} for the respective ambience and siren STFT windows \vec{w}_a and \vec{w}_s is denoted in equation 4:

$$\vec{m} = \begin{cases} \frac{|\vec{w}_s, i|}{|\vec{w}_a, i|} & \text{if } \vec{w}_{a, i} \neq 0 \\ 0 & \text{else} \end{cases} \quad \text{for } i = 1, 2, \dots, 513. \quad (4)$$

Instead of training the network to predict one spectrum mask for one spectrum window as its input, a clustering step provides the network with more time context. The clustering step adds $\frac{N-1}{2}$ windows before and after the STFT window of interest to the network input. The network thus receives packages of magnitude windows in the shape $513 \times N \times 1$. It then outputs a mask in the shape 513×1 for the $\frac{N+1}{2}$ th input window, with all mask entries lying in the interval $[0, 1]$.

Since the network is only fed with the magnitudes of the complex STFT windows, the original phase information of the complex STFT windows is added back to the spectrum windows before the spectral masks are applied. The separated time signal can then be calculated using the inverse short-term Fourier transform.

6. Evaluation

The presented DNN models were evaluated in a quantitative and qualitative manner. The different metrics as well as the separation results from a realistic acoustic scene simulation are presented in the following.

6.1. Network Model Metrics

The final training metrics of the CNN and LSTM models are shown in Figure 5. It is apparent that the CNN with a time context of eight STFT windows converges to consistently lower loss metrics in terms of training, validation and testing compared to the LSTM model. Taking into account the significantly longer training duration of a median of 18199 seconds for the LSTM model compared to the 699 seconds for the CNN model, the CNN clearly achieves better performance.

Since alerts, including sirens, are typically short sounds that stand out from an auditory scene, it was tested if reducing the number of STFT windows fed into the CNN could decrease training and prediction durations without compromising model accuracy. Figure 5 shows training results with eight STFT windows compared to twelve STFT windows.

The CNN with a time context of eight STFT windows shows clear signs of overfitting after about 13 epochs. Conversely, the model with eight windows seems to converge slightly sooner and trains about 25% faster with respect to the median training times of 525 seconds and 699 seconds, respectively.

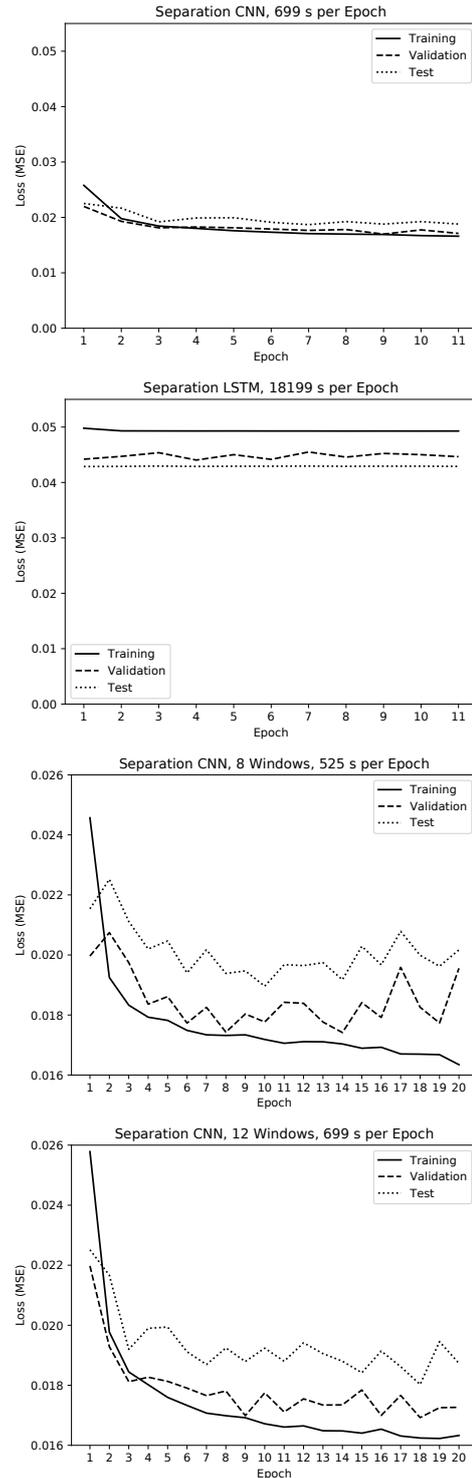


Fig. 5: Training results. The first two charts show the less accurate results of the LSTM model despite a roughly 25 times longer training time compared to the CNN model. The third chart shows the CNN loss trend with a time context of eight STFT windows, the bottom chart for a time context of twelve STFT windows. For each scenario, the median training time for one epoch is denoted above.

After 18 epochs, however, the CNN with a time context of twelve windows reaches a loss minimum below all loss minima of the eight window time context model. The CNN was therefore identified as the better performing model.

6.2. Spatial Audio Scene Simulation

For a realistic evaluation of the system with the better performing CNN model, recordings in a large test room with a three-dimensional speaker arrangement were made. The room dimensions are 9 m × 7.1 m × 4.7 m (L×B×H). The speaker setup consists of 49 satellite speakers and four subwoofers.

The Fraunhofer SpatialSound Wave technology was used to play back three-dimensional ambience recordings on the speaker setup via the object-based audio approach [5]. Besides the realistic playback of the ambience recordings, the setup enables the mapping of a siren signal to an audio object which can arbitrarily be moved in space.

Two condenser microphones were placed at the acoustic hot spot of the demo room. A distance of 0.2 meters between the microphones was chosen for a distance similar to small microphones as they could be attached to the back of ear cups.

Three different sirens were each moved along a circular and linear audio object path as shown in Figure 6. The

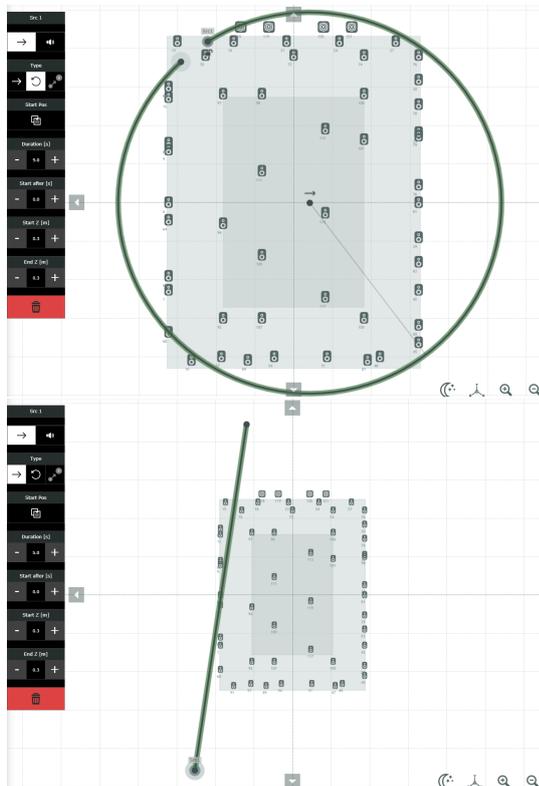


Fig. 6: Automated siren movements. The sirens are moved along the 360° circular path at the top to explicitly test localization accuracy. The linear path at the bottom simulates sirens passing by in traffic.

sirens represent a German, American Wail and American Yelp police siren. Common traffic noise was played back from a three-dimensional recording to simulate a realistic traffic ambience. The volume ratio between the sirens and traffic ambience was adjusted by ear. For the circular siren movement, a constant siren gain was chosen to explicitly test the localization accuracy of the CNN model at every azimuth α . In the case of the linear siren movement, the volume of the siren was automated to be distant dependent, i.e.

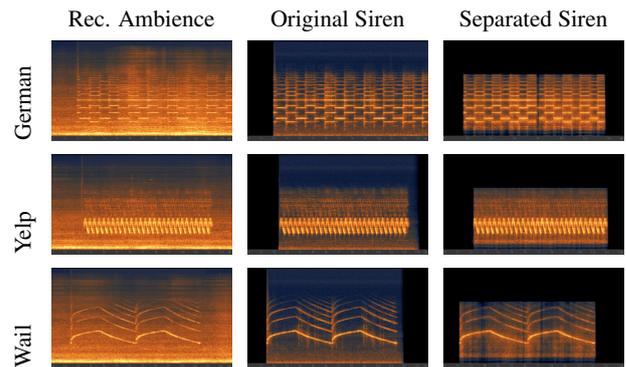


Fig. 7: Separation results for circular siren movements. The CNN produces reliable results apart from short interruptions in the German and wail siren signals.

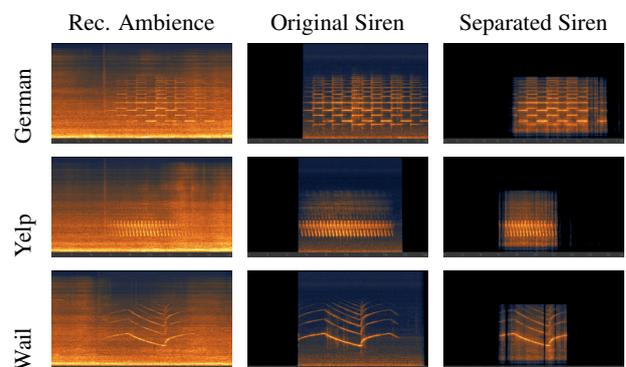


Fig. 8: Separation results for linear siren movements. The louder parts of the sirens are separated successfully, while the quieter parts are not always recognized.

increasing towards the microphone position and decreasing while moving away from it.

Separation results for the circular siren movement are shown in Figure 7. The spectrograms are plotted on the Mel frequency scale using the monophonic sum of the signals [11]. The separated sirens are close to the original siren signals and the sections without sirens are successfully kept quiet by the network. The slightly brighter areas around the main sinusoidal siren sound waves indicate a low remaining noise floor in the separated sequences.

Separation results for the linear siren movement path are plotted in Figure 8. Contrary to the separation results of the constant siren volume on the circular movement path, these results reveal unreliable separations for the quieter siren sections. While the German siren is recognized throughout most of its occurrence in the recorded ambience mix, both the yelp and wail siren are only separated properly during the louder sections, i.e. when the virtual distance of the siren decreases towards the microphone position. This indicates that the CNN is not able to reliably detect sirens below a certain volume threshold.

Although quieter siren signals appear to remain a challenge for the tested models, quiet sirens may also be less of importance if they are far away. Further simulated or real-life

tests will therefore be needed to reliably identify the detection boundaries of the proposed architecture and optimize the presented system and training data accordingly.

7. Conclusion

This paper compared a CNN and LSTM model with a corresponding system architecture for the separation of sirens. The CNN model training is quicker and produces more accurate results. The CNN model was also evaluated from a practical standpoint using the Fraunhofer Spatial SoundWave technology for an object-based traffic scene playback. The circular movement of three typical sirens around the microphone position showed reliable separation results. The linear movement simulation revealed detection boundaries for siren signals below a certain volume threshold.

Future research may reveal more efficient architectures that meet the limited system resources and realtime requirements of an embedded system inside noise-cancelling headphones. Processing delays will need to be kept within strict time constraints, for instance in dangerous traffic situations, while conserving system resources, e.g. with respect to battery life. For a guaranteed reliable performance in difficult situations, the exact boundaries of the proposed system will need to be identified and corresponding training data optimizations will need to be made.

8. References

- [1] Estefania Cano and Hanna Lukashevich. 2019. Selective Hearing: A Machine Listening Perspective. In *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE.
- [2] Pierre Comon and Christian Jutten. 2010. *Handbook of Blind Source Separation: Independent component analysis and applications*. Academic press.
- [3] Mark Daoust. 2019. Keras. <https://www.tensorflow.org/guide/keras>. [Online; accessed 21-August-2019].
- [4] Derry Fitzgerald, Antoine Liutkus, and Roland Badeau. 2016. Projection-based demixing of spatial audio. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)* 24, 9 (2016), 1556–1568.
- [5] Alejandro Gasull Ruiz, Christoph Sladeczek, and Thomas Sporer. 2015. A description of an object-based audio workflow for media productions. In *Audio Engineering Society Conference: 57th International Conference: The Future of Audio Entertainment Technology—Cinema, Television and the Internet*. Audio Engineering Society.
- [6] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
- [7] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).
- [8] Tominori Kimura. 2011. Noise-cancelling headphone. US Patent 8,045,726.
- [9] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [10] Alejandro Koretzky, Karthiek Reddy Bokka, and Naveen Sasalu Rajashekhara. 2018. Real-time audio source separation using deep neural networks. US Patent 10,014,002.
- [11] Beth Logan et al. 2000. Mel Frequency Cepstral Coefficients for Music Modeling.. In *ISMIR*, Vol. 270. ISMIR, 1–11.
- [12] John C Middlebrooks and David M Green. 1991. Sound localization by human listeners. *Annual review of psychology* 42, 1 (1991), 135–159.
- [13] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2012. Understanding the exploding gradient problem. *CoRR, abs/1211.5063* 2 (2012).
- [14] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. 2014. A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 1041–1044.
- [15] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.
- [16] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. 2015. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853* (2015).